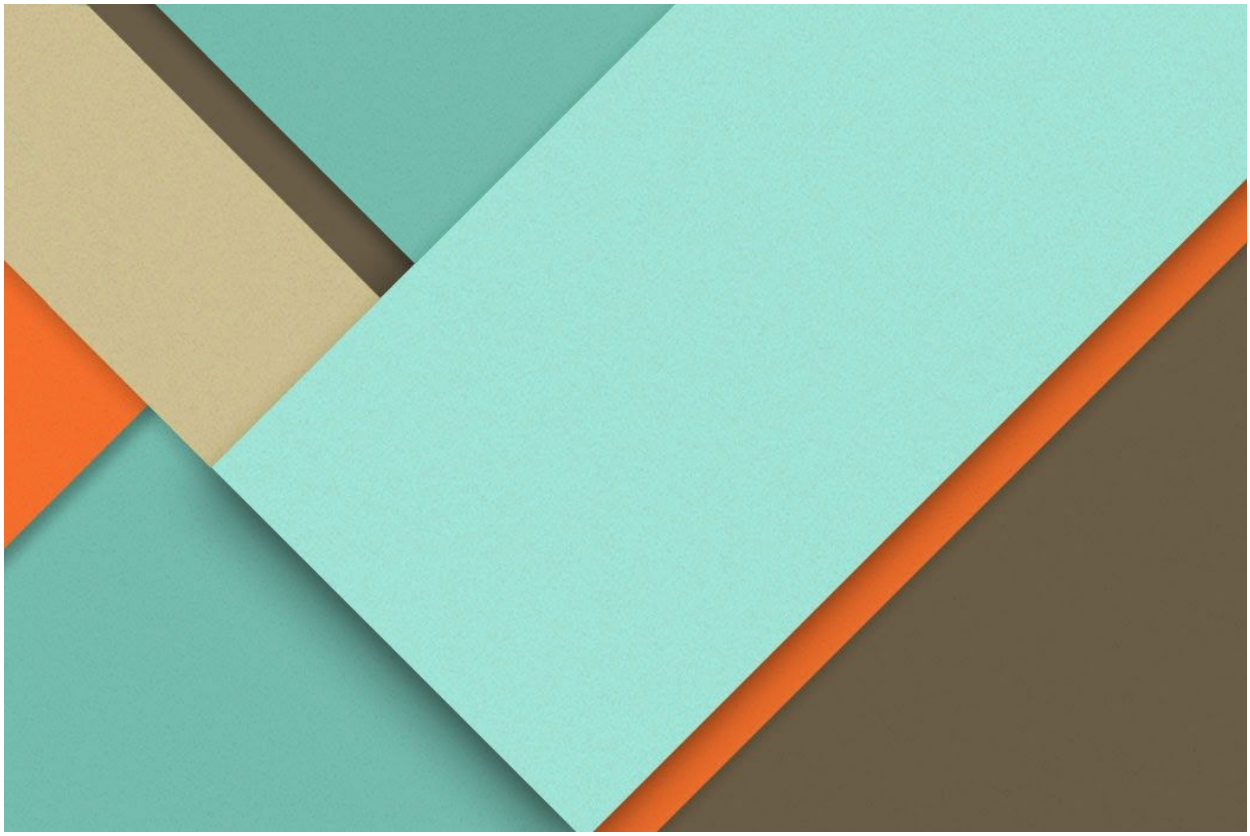
UNIVERSIDAD TECNOLÓGICA NACIONAL - FACULTAD REGIONAL CÓRDOBA



# CO-ALQUILANDO

METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE

26/06/2020

## Release 1


Grupo 1

| | |
|---|---|
| Conti, Bruno | 72890 |
| Lucero Schneider, Pablo | 67041 |
| Peralta, Agustin | 71759 |

Profesores

Ing. Juan Pablo Bruno

Ing. Natalia Andriano

# Índice

## Visión general

En el siguiente documento presentaremos el Release de una parte de la funcionalidad de la aplicación Co-Alquilando, esté mismo fue desarrollado en el marco de la Materia Metodologías Ágiles en el Desarrollo de Software siguiendo las pautas fijadas por el equipo docente y con el objetivo de aplicar e integrar los conocimientos teóricos.

# Sprint 4

## Gestión de Configuración

### Label / Tags

Por cada Sprint trabajado durante el cuatrimestre realizamos un Tag, hasta la fecha se realizaron 4 tags. A continuación el detalle de cada una de ellos:

- Sprint 1: v1.0.0. - Se realizó una primera versión del listado de las propiedades, los filtros y la página Inicial.

  https://github.com/pabloluceroschneider/Grupo1-MADS-2020/releases/tag/v1.0.0

- Sprint 2: v1.1.0 - Se realizó una segunda versión del listado de las propiedades, se incorporó el mapa para visualizar la ubicación de las misma como también las amenities.

  https://github.com/pabloluceroschneider/Grupo1-MADS-2020/releases/tag/v1.1.0

- Sprint 3: v1.2.0 - Se realizó una tercera versión del listado de las propiedades, se mejoraron aspectos en la página Inicial, se incorporó un Carrousell de imagenes para el listado de propiedades y se implementaron filtros no excluyentes.

  https://github.com/pabloluceroschneider/Grupo1-MADS-2020/releases/tag/v1.2.0

### Documentar Pipeline o Investigación

Si bien el equipo no consideró en este release realizar un Pipeline en un servidor de Jenkins, si se investigó y se va a tener en cuenta para futuros releases realizar un proceso de Integración Continua haciendo uso de esta herramienta. La idea es implementar una estrategia de integración continua cada vez que el se realiza un pull request en la branch master, dado que el equipo trabaja las funcionalidades en diferentes ramas a lo largo del desarrollo. De esta forma cada vez que una funcionalidad esté lista y testeada se realiza un pull-request y en este punto se ejecutarán también los test unitarios para garantizar que el build no sea defectuoso.

## Testing

### Trazabilidad

El equipo a lo largo de los sprints, fue trabajando sobre los casos de prueba, en cada sprint se definieron casos para los criterios de aceptación. Cada bug que se detectó se lo clasificó según su severidad y prioridad y se definió como estrategia hacer un seguimiento de aquellos bugs que tuvieran severidad alta y crítica y prioridad alta o bloqueador. Estos bugs iban siendo analizados continuamente a lo largo del desarrollo para garantizar su solución y de esta forma tratar de aumentar la calidad del producto.

### Completitud del test

Se consideró que la fase de testing en cada sprint estuvo terminada cuando el 100% de los casos de pruebas definidas para ese sprint se pasaron con éxito y también cuando en un sprint se encontraron y se repararon más de 4 bugs sobre una determinada funcionalidad.

### Reporte de Pruebas

A lo largo de los sprint del release el equipo encontró:

- El listado de pruebas no fue diseñado mediante un enfoque responsive
- Ausencia de formato de fecha en el listado de propiedades.
- Filtros Excluyentes
- Las opciones de la página inicial no tienen un diseño responsive
- El mapa de ubicación de propiedades no tiene un diseño responsive
- No se pueden visualizar más de 8 amenities

## Bugs Conocidos

- La lista de opciones en la página inicial no está diseñada Responsive.
- El mapa donde se visualiza la ubicación de la propiedad no está diseñada Responsive.
- El filtrado se realiza a nivel de front-end
- La aplicación no tiene paginación, por lo que implica posibles bugs en pruebas de carga o rendimiento.

## Automatizar Test o Documentar Investigación

En el sprint 3 el equipo decidió incorporar la automatización de test en el servidor de Back-end, se llevaron a cabo 3 test automatizados, utilizando el Framework de pruebas Jest para javaScripts. El equipo para los siguientes releases tiene pensado aumentar la cantidad de test automatizados en el Back-End como también incorporar la automatización de test funcionales mediante la herramienta selenium.

## Test No Funcionales

En este release no se implementarán test no funcionales, ya que el equipo no posee conocimientos en este campo. Pero se evalúa la incorporación de estos test en los siguientes Releases.

# Documentos

## Notas de Release

El Release 1 incluye las siguientes funcionalidades:

- **Página inicial**
  - Se implementó una página Home que soporta el todo el contenido del proyecto: opciones de filtro principales y listado de propiedades.
- **Funcionalidades para propiedades**
  - Los usuarios pueden visualizar las propiedades en formato de lista, con las características principales de la misma. Esto incluye: Título, precio, amenities, 5 fotos (carrousel), tag de dueño/inmobiliaria y un mapa con la ubicación aproximada.
  - Los usuarios pueden agregar una nueva propiedad al listado.
- **Funcionalidades para filtrado de propiedades**
  - Los usuarios pueden filtrar las propiedades de manera no excluyente según: dueño o inmobiliario, cantidad de habitaciones y/o amenities de la misma.

## Identificación del build, despliegues y releases.

Para gestionar releases, despliegues, builds y correcciones, se utilizarán tags utilizando las siguientes reglas:

Para la identificación de releases utilizaremos el primer dígito:

tag v1.0.0.0

tag v2.0.0.0

Para la identificación de despliegues utilizaremos el segundo dígito:

tag v0.1.0.0

tag v0.2.0.0

...

Habrá builds que necesitemos identificar como tag pero no desplegar, y para ello utilizaremos el tercer dígito del tag:

tag v0.1.1.0

tag v0.1.2.0

...

Para la identificación de correcciones utilizaremos el cuarto dígito:

tag v0.1.1.1

tag v0.1.1.2

...

En el nombre de cada tag se especificará en qué Sprint fue desarrollado. Además, una breve descripción de los cambios con respecto al anterior tag.

## Prácticas Continuas.

Para las automatizaciones de builds y despliegues, se continuará con las herramientas e investigaciones llevadas a cabo hasta el momento.

Para los test automatizados, se incrementarán los test unitarios desarrollados en jest. Hasta el momento se han hecho 3 test, cuya funcionalidad se reduce a probar el

funcionamiento del método que realiza el filtrado de publicaciones. Lo próximo será testear el renderizado mediante snapshots, los request http, y más funcionalidades.

En cuanto a la herramienta para automatizar el despliegue, el plan consiste en comenzar a utilizar Jenkins. A continuación, se dejará una constancia de lo investigado.

- Crear el archivo Jenkinsfile y configurarlo:

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                sh 'npm install'
            }
        }
        stage('Test') {
            steps {
                sh 'npm run test'
            }
        }
    }

}
```

agent any: Esto significa que cualquier agente disponible puede ejecutar las etapas definidas en el pipeline.

stages: En los stages definimos las etapas.

En el step de Build nos descargamos las dependencias, y en el de Test, ejecutamos el comando que corre los test de Jest.